

## We are the makers – Scénario d'apprentissage **IoT**

Auteur: Chrissa Papasarantou

Titre	<i><b>Créer un système d'alarme</b></i>
<p><b>1. scénario</b></p>	<p>La protection de la statuette Oscar vous a été confiée par le comité des Oscars (Figure 1). Par conséquent, vous avez décidé de créer un système d'alarme qui vous avertira au cas où quelqu'un tenterait de le voler. Ainsi, votre objectif principal pour cette activité est de créer un système basé sur Arduino, équipé de capteurs et de composants électroniques appropriés, qui vous avertira de plusieurs manières (c'est-à-dire avec des signaux audibles et optiques) lorsque la statuette est retirée. Outre la création et la programmation du circuit, les étudiants doivent être encouragés à donner un peu de vie au projet à travers l'artisanat. Le développement d'idées supplémentaires (par exemple, la création d'un système qui surveillera les activités enregistrées du système d'alarme et informera l'utilisateur à distance) doit également être encouragé.</p> 
<p><b>2. Groupe cible</b></p>	<p>Ce scénario peut être adapté aux âges: 12-15 ans</p>
<p><b>3. Durée</b></p>	<p>Ce scénario peut être mis en œuvre en classe en 2 sessions (2-3 heures chacune)</p>
<p><b>4. Besoins couverts par l'exercice</b></p>	<ul style="list-style-type: none"> <li>- Compréhension de la théorie Arduino de base (modules, add-ons, plateforme, langage de programmation, etc.)</li> <li>- Compréhension du fonctionnement des capteurs</li> <li>- Mise en évidence des méthodes de mise en œuvre et d'intégration de systèmes informatiques à une échelle plutôt petite (c'est-à-dire à domicile).</li> </ul>
<p><b>5. Résultats attendus</b></p>	<ul style="list-style-type: none"> <li>- Réalisation de montages Arduino de base</li> <li>- Utilisation efficace de la programmation par blocs pour les projets de base</li> <li>- Programmation Arduino de base (code)</li> <li>- Utilisation et programmation efficaces avec des capteurs</li> </ul>
<p><b>6. Methodologies</b></p>	<p><b>Leçon 1 : Session de bienvenue</b></p> <ul style="list-style-type: none"> <li>- Formation des équipes</li> <li>- Brève introduction / présentation: Présentation du scénario et des objectifs du projet, définition des objectifs de l'équipe, élaboration du résultat final / résultat - Arduino: Première familiarisation</li> </ul> <p><b>Leçon 2 :</b></p> <ul style="list-style-type: none"> <li>- Construction Arduino (cartes, capteurs, etc.)</li> <li>- mBlocks: commandes, compilation, exécution</li> </ul>

	<ul style="list-style-type: none"> <li>- Code Arduino: un ensemble de commandes est introduit, et une explication est fournie</li> </ul> <p><b>Leçon 3 :</b></p> <ul style="list-style-type: none"> <li>- Programmation vers l'implémentation de tâches (mBlock ou Arduino IDE). Il est à noter que des solutions à moitié cuites sont également utilisées afin d'impliquer en douceur les étudiants dans la programmation avec mBlock.</li> </ul>
<p><b>7. Lieu / Environnement</b></p>	<p>Laboratoire d'informatique</p>
<p><b>8. Outils / matériaux / ressources</b></p>	<p>Projecteur, système audio, kits Arduino, capteurs</p>
<p><b>9. Description étape par étape de l'activité</b></p>	<p><b>Leçon 1</b></p> <ol style="list-style-type: none"> <li>1. Activité de formation d'une petite équipe - création de liens d'équipe</li> <li>2. Démonstration de courtes vidéos sur les systèmes et méthodes de sécurité (immerger les élèves dans le contexte de l'activité et leur fournir des informations de base).</li> <li>3. Présentation des étapes qui seront suivies pour atteindre les objectifs du projet</li> <li>4. Introduction à Arduino - brève démonstration (par vidéo et / ou démonstration en temps réel)</li> </ol> <p><b>Leçon 2</b></p> <ol style="list-style-type: none"> <li>1. Construction d'Arduino en équipes (fixation cartes / capteurs, etc.)</li> <li>2. Démonstration de mBlock - facile à démarrer avec des tâches à des fins de familiarisation (LED clignotante, etc.)</li> <li>3. Démonstration de la plate-forme de codage Arduino - facile à démarrer avec des tâches de programmation à des fins de familiarization</li> </ol> <p><b>Leçon 3</b></p> <ol style="list-style-type: none"> <li>1. mBlock et / ou plateforme de codage pour mettre en œuvre le projet (création d'un système d'alarme)</li> <li>2. Tester les solutions</li> </ol> <p>Discussion - conclusions Ce projet est-il lié à la vie réelle ? Tient-il compte des risques réels ?</p>
<p><b>10. Retour d'information</b></p>	<p><b>Leçon 1 :</b> Par la discussion, l'enseignant décide si les élèves ont réalisé l'intérêt de transformer les objets de la vie quotidienne en objets intelligents.</p> <p><b>Leçon 2 :</b> Le montant du succès des petits projets (construction et programmation)</p> <p><b>Leçon 3 :</b> Focus sur la contribution de chaque équipe à la réalisation du projet</p>

## 11. Evaluation

**Leçon 1 :** Un court questionnaire est donné à remplir aux étudiants. Le questionnaire se concentre sur le sujet du projet et vise à explorer les perceptions des étudiants sur des sujets liés à la mise en œuvre de systèmes informatiques à petite échelle.

**Leçon 2 :** Des groupes de discussion sont organisés afin d'explorer comment chaque équipe a travaillé pour atteindre l'objectif final, la dynamique de l'équipe et la manière dont les tâches ont été effectuées et les échecs ont été rencontrés.

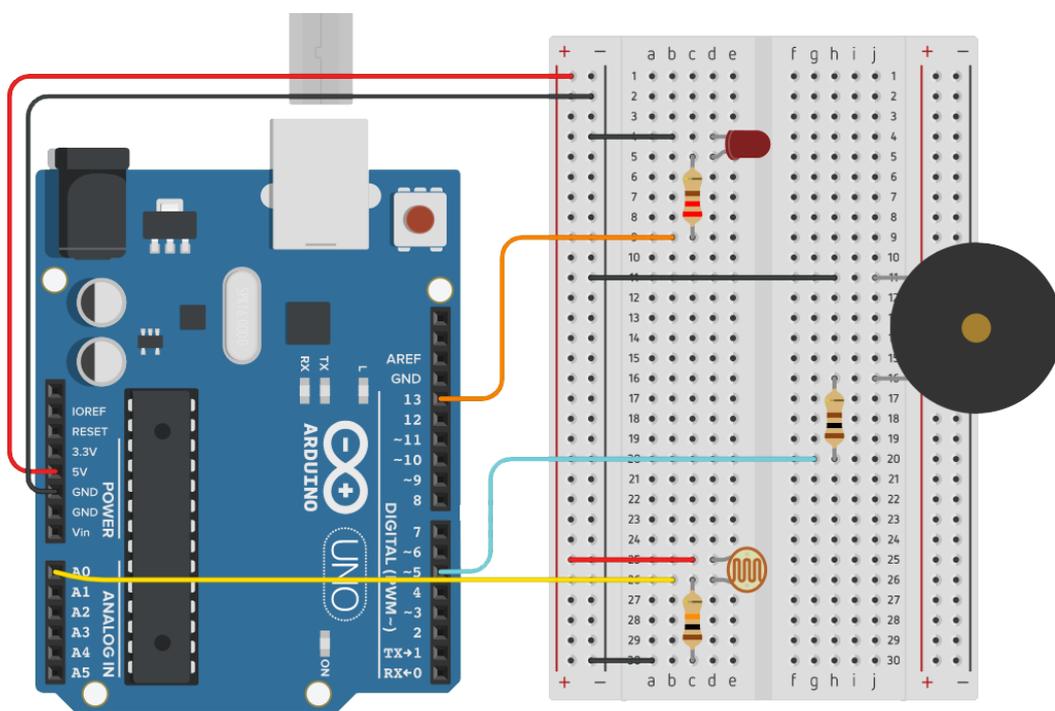
**Leçon 3 :** Le projet final est évalué d'un point de vue technique et conceptuel. Il est intéressant de voir quel type d'outils les étudiants ont utilisé et mélangé, la complexité des solutions qu'ils ont mises en œuvre, si le scénario du projet a été étendu, si des idées de solutions optimales ont été avancées. L'évaluation est basée sur des observations en cours pendant la mise en œuvre du projet et sur l'examen du résultat final (par l'enseignant).

## Description du projet

**Concept** : Ce projet concerne la création et le système d'alarme: lorsque quelqu'un essaie de retirer un objet, les signaux audio et optiques sont activés.

### Création du circuit :

Le schéma suivant (*Figure 2*) présente la manière dont les composants du circuit, c'est-à-dire une LED, un buzzer et une photorésistance, doivent être connectés. Au début, la maquette doit être alimentée (5V) et la masse (GND), via des cavaliers qui sont respectivement connectés aux colonnes + et - de la maquette. L'anode de la LED (jambe plus longue) est connectée à l'une des broches numériques (13 dans l'exemple) via une résistance de 220  $\Omega$ , tandis que la cathode (jambe plus courte) est connectée à la masse. L'une des broches du buzzer est connectée à l'une des broches numériques PWM (5 dans l'exemple) via une résistance de 100  $\Omega$ , tandis que l'autre broche est connectée à la masse. Enfin, l'une des broches de la photorésistance est connectée à l'alimentation (5V), tandis que l'autre est connectée à l'une des broches analogiques (A0 dans l'exemple) ainsi qu'à la masse, via une résistance de 10K $\Omega$ . L'ensemble du système peut également être alimenté par des banques solaires.



*Figure 1: Diagramme du circuit*

### Programmation du circuit :

La prochaine étape consiste à donner un peu de vie au projet grâce à la programmation. Selon le scénario, lorsque quelqu'un tente de retirer la

statuette, le buzzer et la LED sont activés à travers la photorésistance. Par conséquent, un niveau de lumière ambiante, sur lequel la photorésistance activera le reste des composants du circuit, doit être déterminé. Au-dessus de ce niveau, le buzzer devrait commencer à bourdonner tandis que la LED devrait commencer à clignoter assez rapidement. Le volume doit être assez élevé pour que le son soit facilement perceptible.

Le script suivant (*Figure 3*) est une solution de programmation indicative, créée dans un logiciel de programmation par blocs (mBlock). Selon ce script, un niveau de lumière ambiante (c'est-à-dire 300) est défini afin d'agir comme un point de déclenchement. Au-dessus de ce niveau, la photorésistance ordonne simultanément au buzzer et à la LED de s'activer. Si la lumière ambiante est inférieure à ce niveau, alors les deux composants précités sont désactivés.

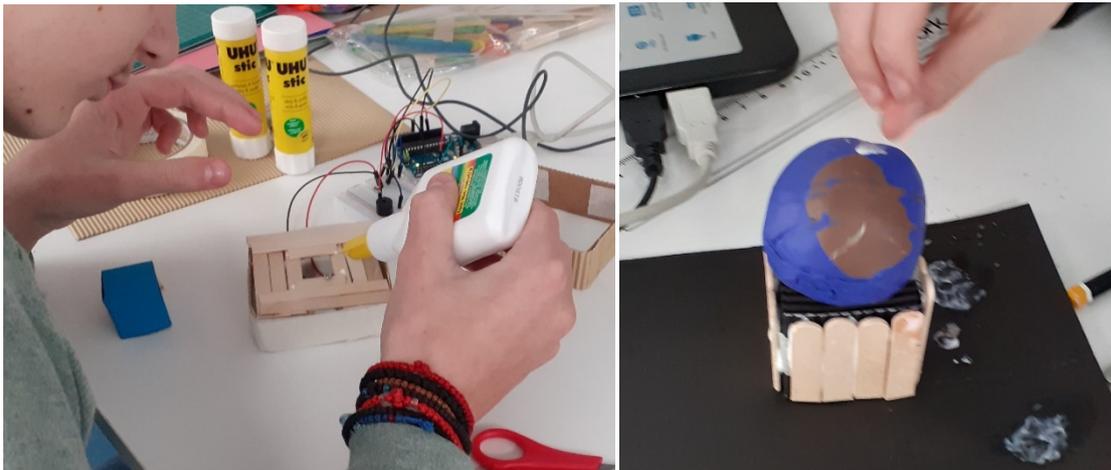
**Conseil:** pour présenter correctement cette activité à votre classe, vous êtes encouragé à fournir des solutions à moitié préparées du script (c'est-à-dire tous les blocs séparément, une version semi-structurée du script, etc.).

```

when Arduino Uno starts up
  forever
    if read analog pin (A) 0 > 300 then
      set PWM 5 output as 300
      set digital pin 13 output as high
      wait 0.2 seconds
      set digital pin 13 output as low
      wait 0.2 seconds
    else
      set digital pin 13 output as low
      set PWM 5 output as 0
  
```

*Figure 2: Script indicatif pour la programmation du système d'alarme*

## Créer un modèle pour représenter le système d'alarme :



Comme cela a déjà été mentionné, les étudiants devraient également être encouragés à créer un modèle du système d'alarme et à intégrer des parties du circuit dans la structure. Ils peuvent utiliser des matériaux facilement accessibles et / ou recyclables (cartons, pâte à modeler, sucettes glacées etc.) pour leur modèle, ou concevoir un modèle 3D qui sera imprimé sur une imprimante 3D.